



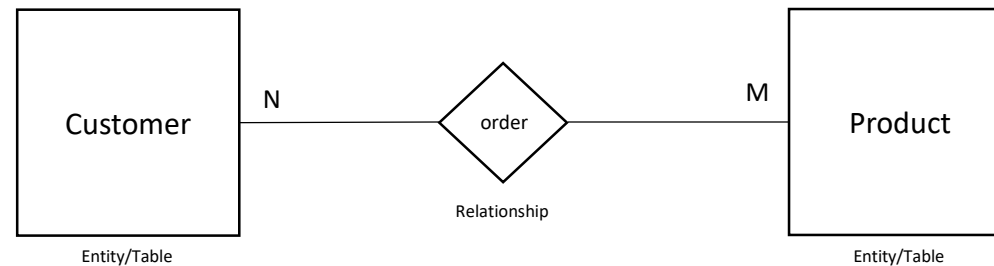
ASP.NET CORE WEB API

Från ERD till en SQL databas

ERD – Entity Relationship Diagram

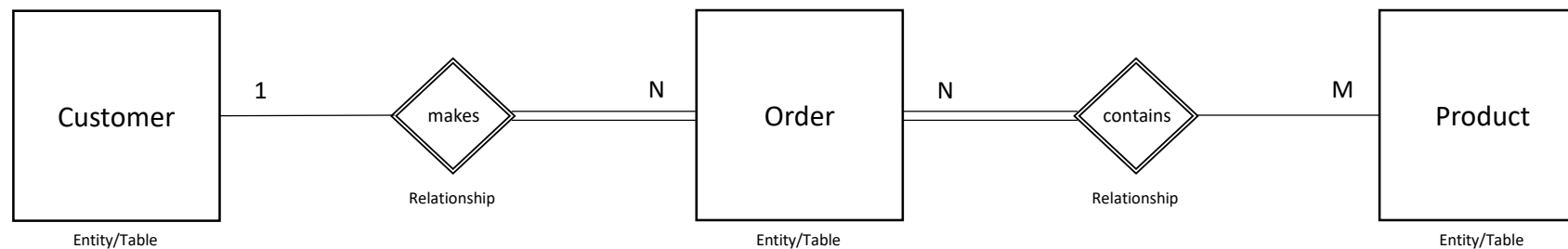
Konceptuell modellering dvs Brainstorming

A Customer can order one or more Products.
A Product can be ordered by one or more Customers



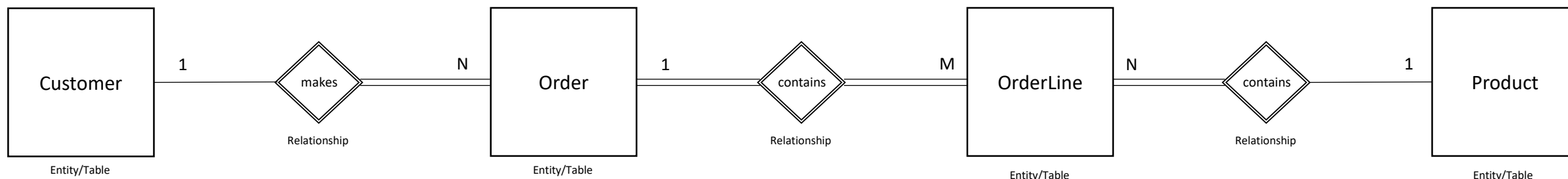
Detta är en många till många relation mellan två entiteter/tabeller. Så kan det inte vara i en databas.
Utan då måste man bygga det med en kopplingstabell emellan.

A **Customer** can make one or more **Orders** and a **Order** must be done by a **Customer**.
A **Product** can be a part of an **Order** and an **Order** must contain one or more **Products**

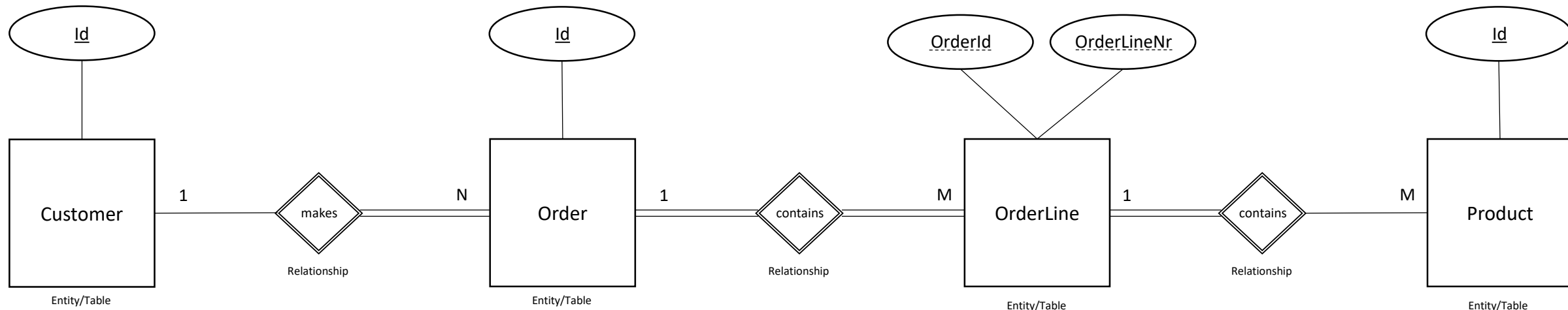


Nu är det relationen en-till många ena sidan. Detta gör att när man lägger en order så måste (som skrivs med dubbla stäck och samma med relationen) det finnas en kund och en produkt kopplad till själva ordern. Det är detta som kallas för en Foreign Key (FK). FK relaterar till en PK i en annan tabell. PK är den Primary Key, det vill säga den som unikt identifierar varje rad i just den tabellen. Men fortfarande har vi problem med relationen mellan Order och Product eftersom en order kan ha en eller flera produkter kopplade till sig.

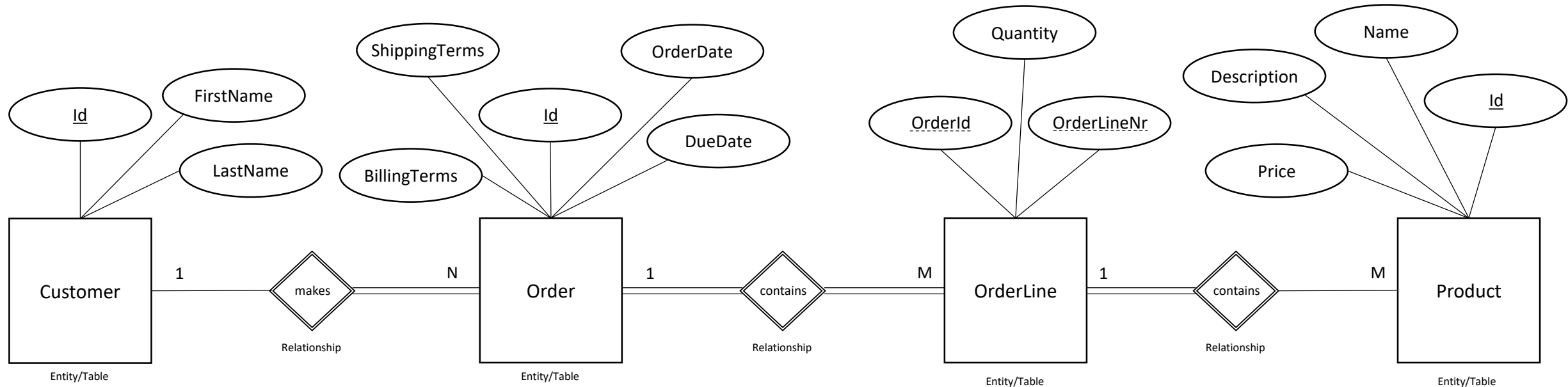
A **Customer** can make one or more **Orders** and a **Order** must be done by a **Customer**.
One **Order** must has one or more **OrderLines** and one **OrderLine** must connect to an **Order**.
A **Product** can be a part of an **OrderLine** and an **OrderLine** must contain one **Products**



Nu är hela kedjan klar. Här har vi lagt till en OrderLine som kommer tala om information om varje rad på en order.



Varje entitet/tabell måste ha något som unikt identifierar varje rad. Detta kan göras med hjälp av en primär nyckel (PK) eller en sammansatt nyckel, Det vill säga en primär nyckel som består av flera attribut. En primär nyckel skrivs med ett sträck under namnet. En sammansatt nyckel skrivs med En sträckad linje under namnet.



Därefter lägger man till alla attribut man kan komma på för en viss entitet/tabell. Sen ska man köra efter Normaliseringsreglerna 1NF, 2NF, 3NF

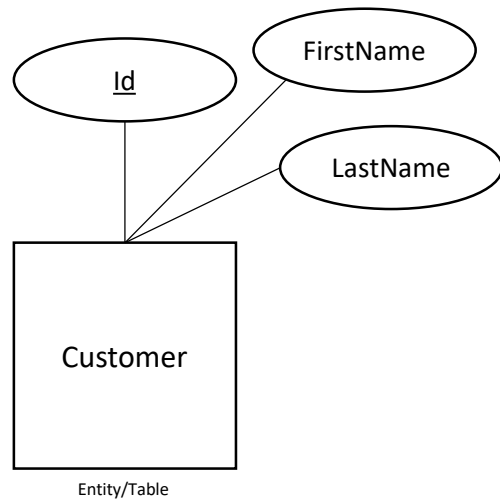
1NF = Alla tabeller ska ha en primär nyckel och det får inte finnas kolumner med flera olika värden i sig. (Såsom Name = Tommy Mattin-Lassei)

2NF = Det får inte finnas repeterande kolumner såsom Product1, Product2, Product3 för vad händer om det kommer en Product4?

3NF = Alla kolumner måste beskriva objektet/tabellen/entiteten. Som telefonnummer är ingen kund, alltså är egentligen ett telefonnummer en egen entitet.

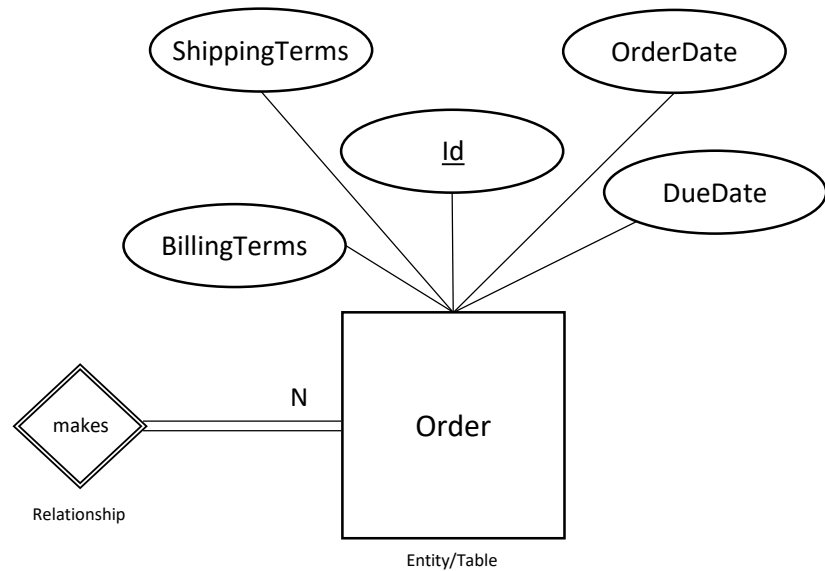
ER-Modellering

Modellering för databaser



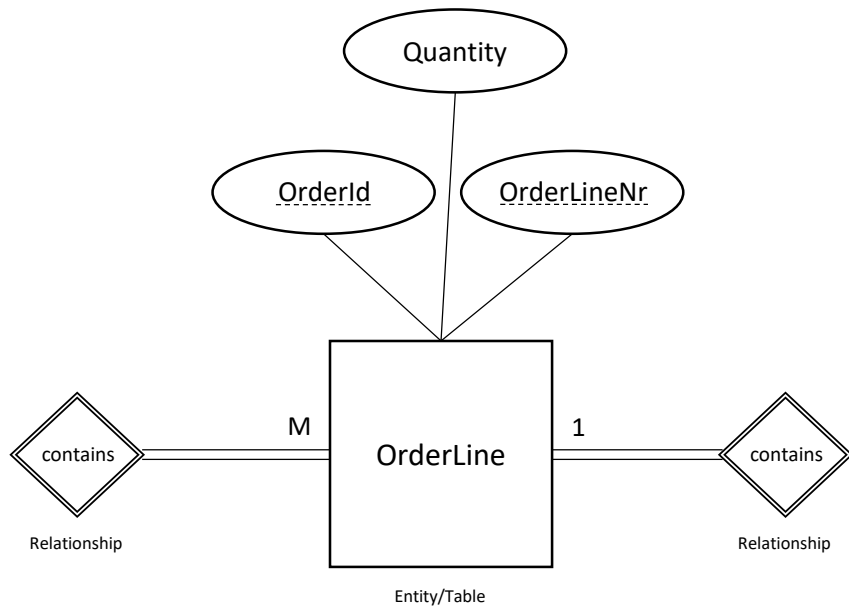
Customer			
PK	Id	int*	not null
	FirstName	nvarchar(50)	not null
	FirstName	nvarchar(50)	not null

Entiteten görs om till att bli en tabell som vi sedan kan skapa upp i självaste databasen. Här talar man om exakt vad tabellen ska heta, vilken är den Primära nyckeln, främmande nycklar, vad för datatyp det ska vara, samt om det är ett tvingande värde eller inte.



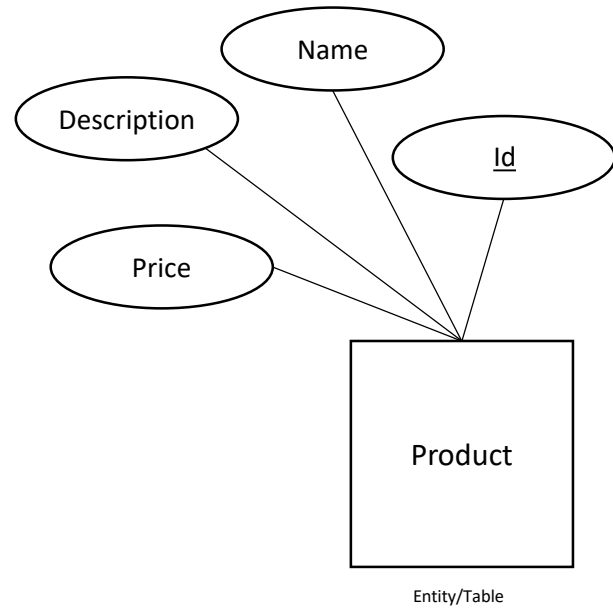
Order			
PK	Id	int*	not null
FK	CustomerId	int	not null
	OrderDate	date	not null
	DueDate	date	not null
	BillingTerms	nvarchar(50)	not null
	ShippingTerms	nvarchar(50)	not null

Entiteten görs om till att bli en tabell som vi sedan kan skapa upp i självaste databasen. Här talar man om exakt vad tabellen ska heta, vilken är den Primära nyckeln, främmande nycklar, vad för datatyp det ska vara, samt om det är ett tvingande värde eller inte.



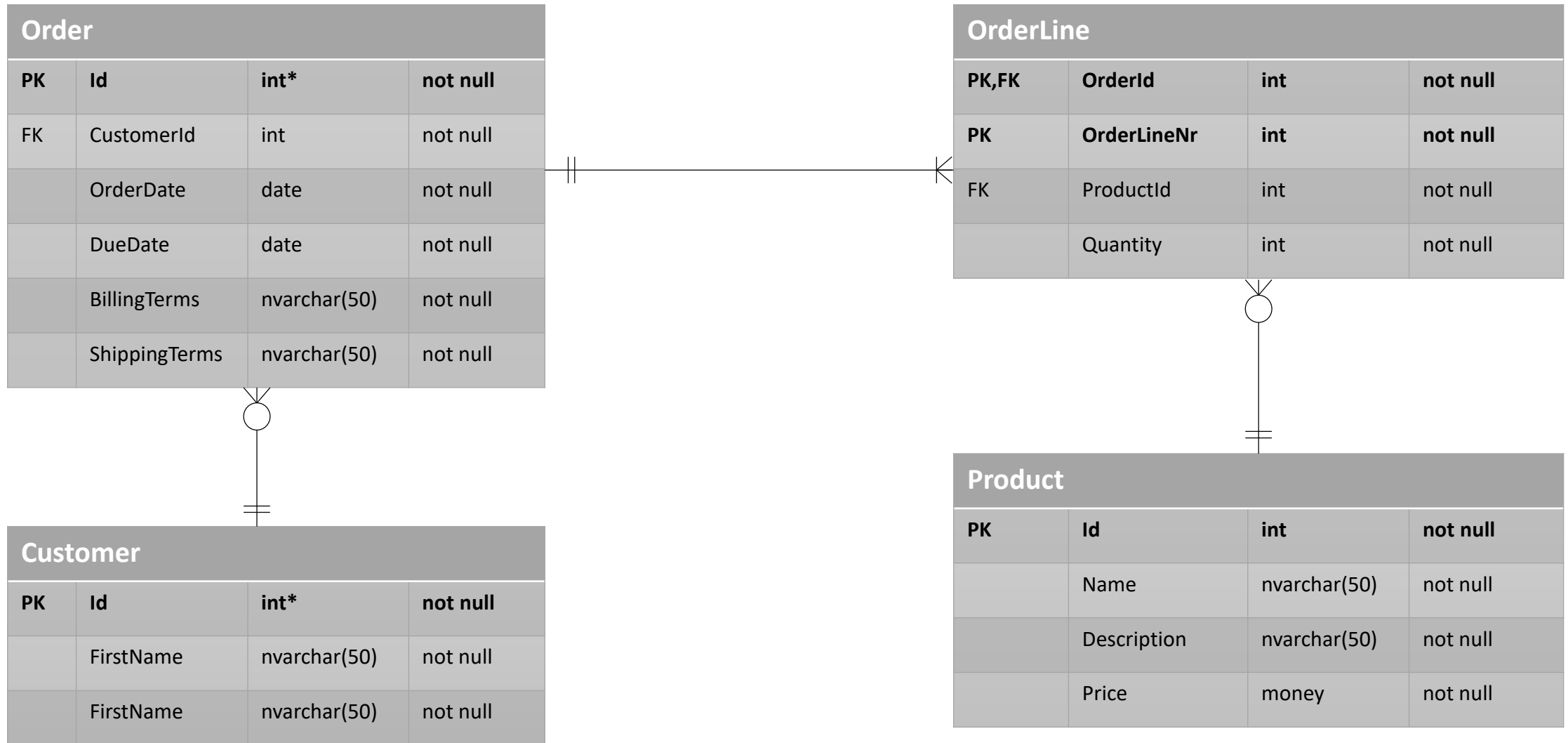
OrderLine			
PK,FK	OrderId	int	not null
PK	OrderLineNr	int	not null
FK	ProductId	int	not null
	Quantity	int	not null

Entiteten görs om till att bli en tabell som vi sedan kan skapa upp i självaste databasen. Här talar man om exakt vad tabellen ska heta, vilken är den Primära nyckeln, främmande nycklar, vad för datatyp det ska vara, samt om det är ett tvingande värde eller inte.



Product			
PK	Id	int	not null
	Name	nvarchar(50)	not null
	Description	nvarchar(50)	not null
	Price	money	not null

Entiteten görs om till att bli en tabell som vi sedan kan skapa upp i självaste databasen. Här talar man om exakt vad tabellen ska heta, vilken är den Primära nyckeln, främmande nycklar, vad för datatyp det ska vara, samt om det är ett tvingande värde eller inte.



Vår färdiga ER-Modellering ser alltså ut såhär. Utifrån denna kan vi skapa vår databas.

SQL Databas

Skapa SQL-syntaxer och SQL-databas lokalt

Customer			
PK	Id	int*	not null
	FirstName	nvarchar(50)	not null
	FirstName	nvarchar(50)	not null

```
CREATE TABLE Customer (
    Id int not null identity(1,1) primary key,
    FirstName nvarchar(50) not null,
    LastName nvarchar(50) not null
)
```

Product			
PK	Id	int	not null
	Name	nvarchar(50)	not null
	Description	nvarchar(50)	not null
	Price	money	not null

```
CREATE TABLE Product (
    Id int not null identity(1,1) primary key,
    Name nvarchar(50) not null,
    Description nvarchar(50) not null,
    Price money not null
)
```

GO

Vi skapar alla tabeller som inte har några FK först. Annars kommer det bli fel. Vi skapar dem först genom att säga GO

Order			
PK	Id	int*	not null
FK	CustomerId	int	not null
	OrderDate	date	not null
	DueDate	date	not null
	BillingTerms	nvarchar(50)	not null
	ShippingTerms	nvarchar(50)	not null

```

CREATE TABLE [Order] (
    Id int not null identity(1,1) primary key,
    CustomerId int not null references Customer(Id),
    OrderDate date not null,
    DueDate date not null,
    BillingTerms nvarchar(50) not null,
    ShippingTerms nvarchar(50) not null
)

GO

```

Därefter skapar vi tabeller som har beroende av de första tabellerna. Det görs genom constraints - references

OrderLine			
PK,FK	OrderId	int	not null
PK	OrderLineNr	int	not null
FK	ProductId	int	not null
	Quantity	int	not null

```
CREATE TABLE OrderLine (
    OrderId int not null references [Order](Id),
    OrderLineNr int not null,
    ProductId int not null references Product(Id),
    Quantity int not null

    CONSTRAINT PK_OrderLine
        PRIMARY KEY (OrderId, OrderLineNr)
)
```

Sist men inte minst tar vi den sista tabellen som har flest beroenden i sig. Som endast går att skapa när alla andra tabeller är skapade.



Visual Studio 2019

Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

Get started



Clone or check out code

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

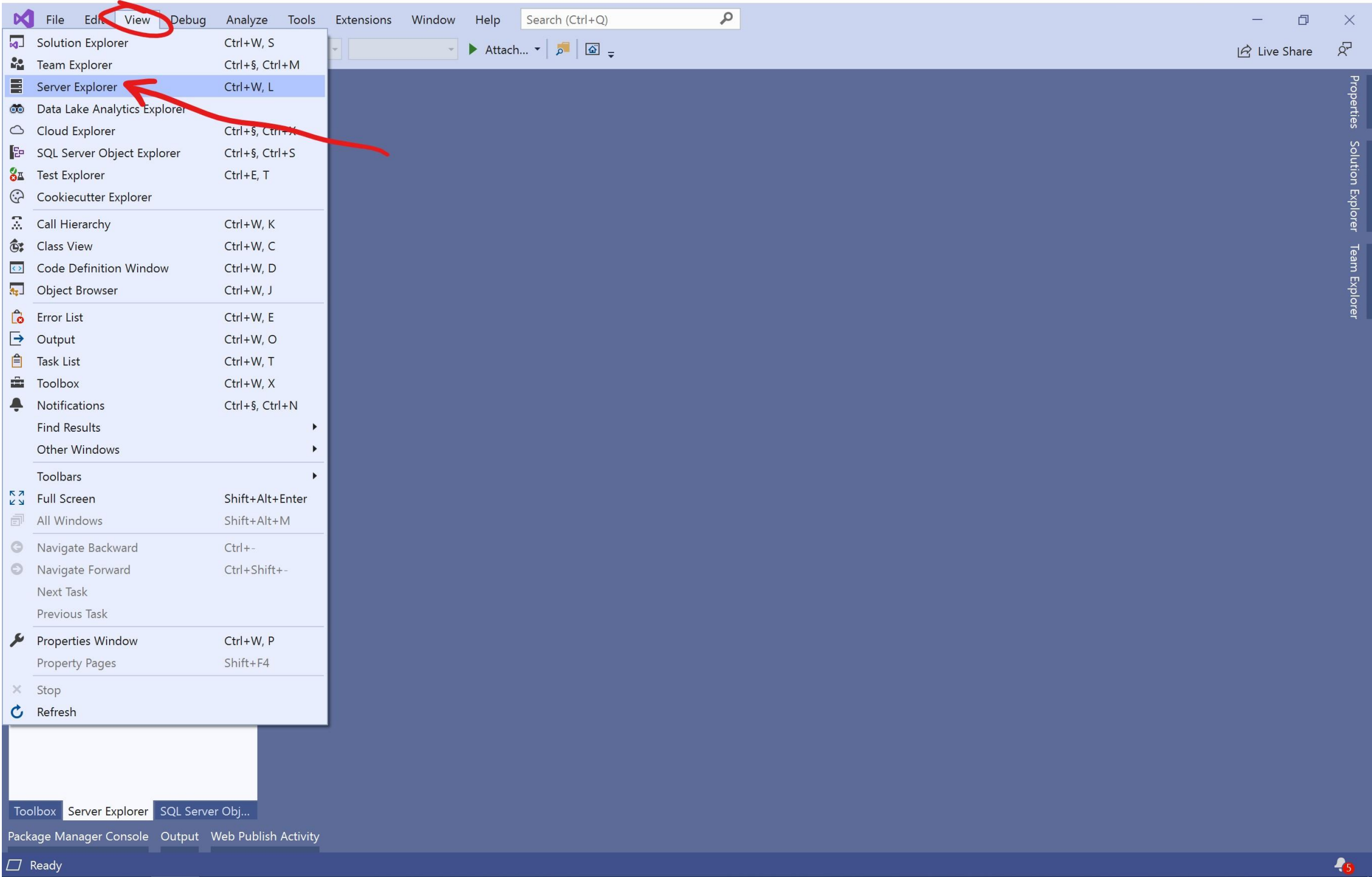
Navigate and edit code within any folder

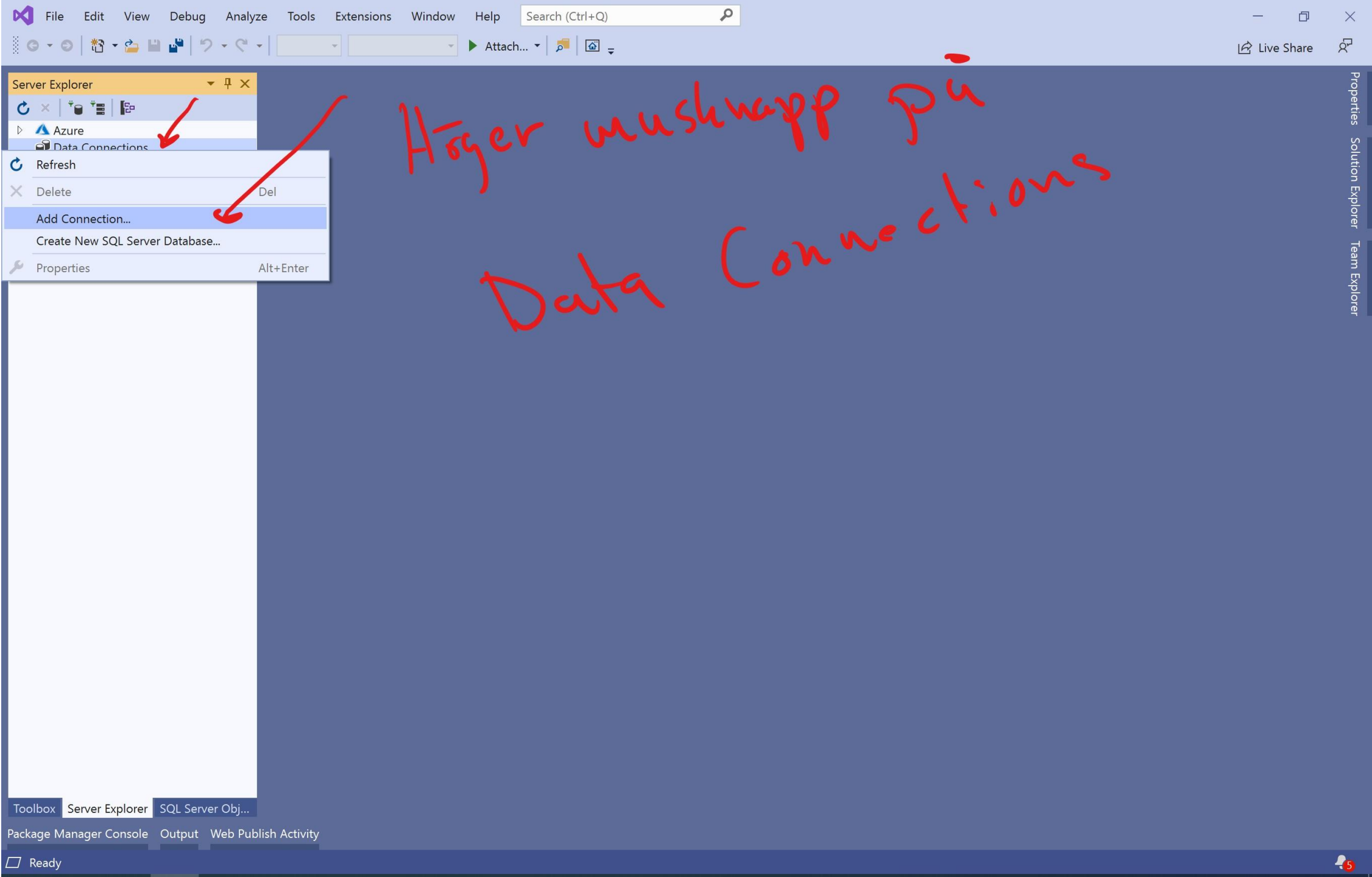


Create a new project

Choose a project template with code scaffolding to get started

[Continue without code →](#)





Server Explorer

- Azure
- Data Connections
- Servers

Change Data Source

Data source:

Microsoft Access Database File

Microsoft ODBC Data Source

Microsoft SQL Server

Microsoft SQL Server Database File

Oracle Database

<other>

Description

Use this selection to attach a database file to a local Microsoft SQL Server instance (including Microsoft SQL Express) using the .NET Framework Data Provider for SQL Server.

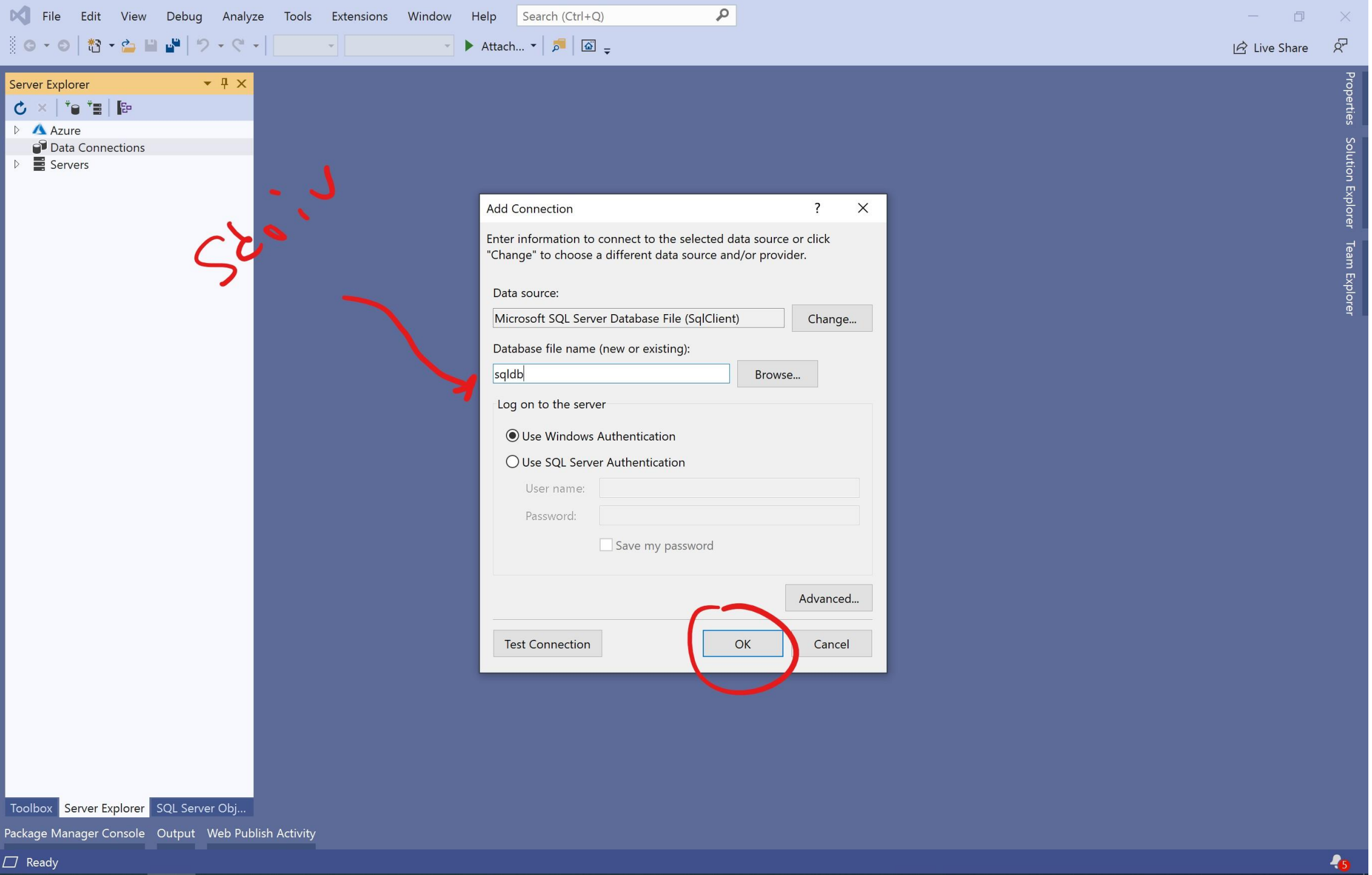
Data provider:

.NET Framework Data Provider for SC

☐ Always use this selection

OK

Cancel



Server Explorer

- Azure
- Data Connections
- Servers

Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:

Microsoft SQL Server Database File (SqlClient) Change...

Database file name (new or existing):

Microsoft Visual Studio

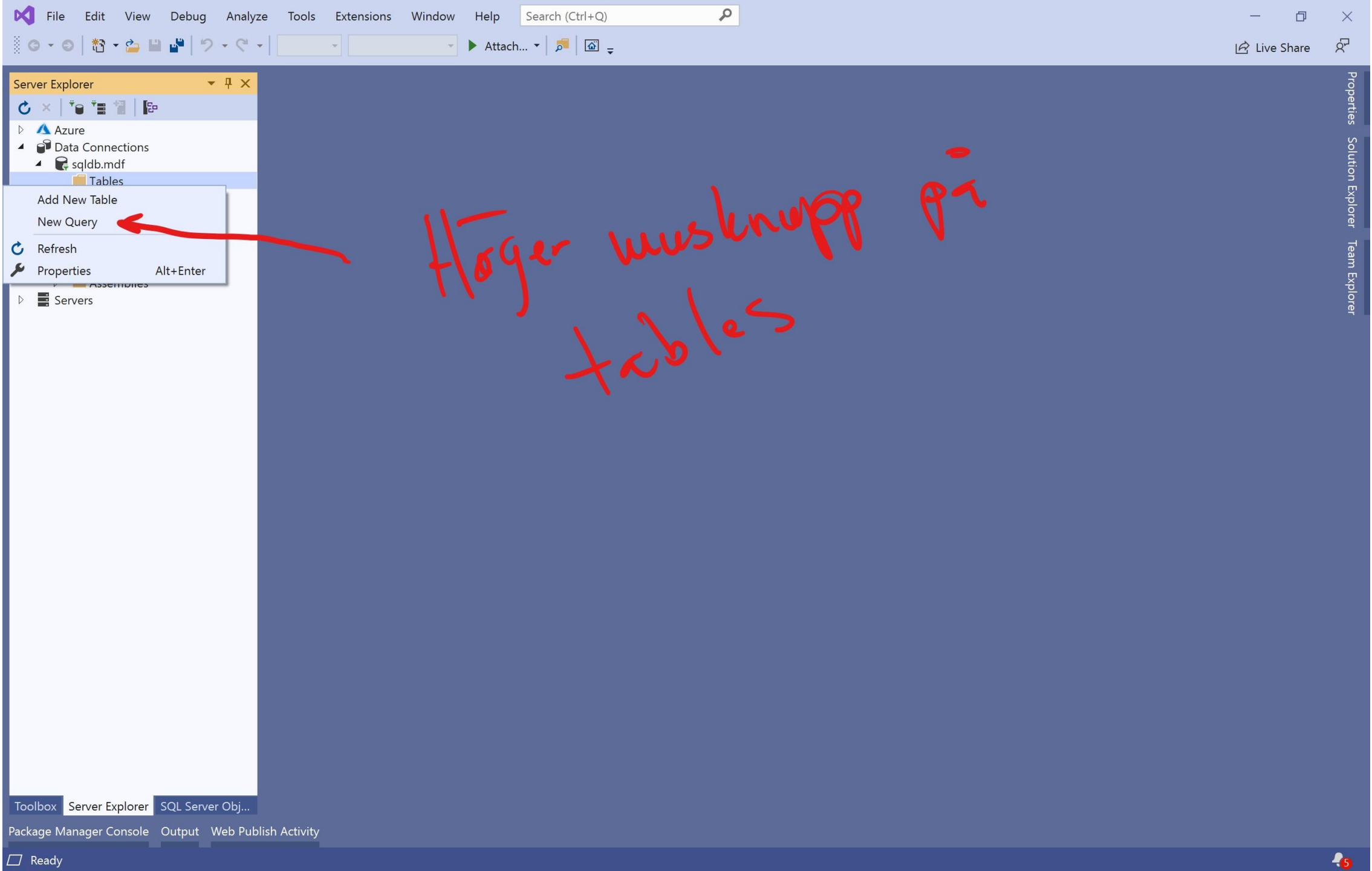
The database file `\sqldb.mdf` does not exist.

Would you like to create it?

Yes No

Advanced...

Test Connection OK Cancel



File Edit View Project Build Debug SQL Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Solution1

Live Share

Server Explorer

Azure

Data Connections

sqlldb.mdf

Tables

Views

Stored Procedures

Functions

Synonyms

Types

Assemblies

Servers

Execute (Ctrl+Shift+E)

1 CREATE TABLE Customer (
2 Id int not null identity(1,1) primary key,
3 FirstName nvarchar(50) not null,
4 LastName nvarchar(50) not null
5)
6
7 CREATE TABLE Product (
8 Id int not null identity(1,1) primary key,
9 Name nvarchar(50) not null,
10 Description nvarchar(50) not null,
11 Price money not null
12)
13 GO
14
15 CREATE TABLE [Order] (
16 Id int not null identity(1,1) primary key,
17 CustomerId int not null references Customer(Id),
18 OrderDate date not null,
19 DueDate date not null,
20 BillingTerms nvarchar(50) not null,
21 ShippingTerms nvarchar(50) not null
22)
23 GO
24
25 CREATE TABLE OrderLine (
26 OrderId int not null references [Order](Id),
27 OrderLineNr int not null,
28 ProductId int not null references Product(Id),
29 Quantity int not null
30
31 CONSTRAINT PK_OrderLine PRIMARY KEY (OrderId, OrderLineNr)
32)

117 % No issues found Ln: 32 Ch: 2 TABS CRLF

00:00:00 0 rows

Ready Add to Source Control

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Solution1

Server Explorer Azure Data Connections Refresh Delete Del Change View Modify Connection... Close Connection New Query Detach Database Browse In SQL Server Object Explorer Rename Properties Alt+Enter

SQLQuery1.sql * SQLQuery1.sql

```
1 CREATE TABLE Customer (  
2     Id int not null identity(1,1) primary key,  
3     FirstName nvarchar(50) not null,  
4     LastName nvarchar(50) not null  
5 )  
6  
7 CREATE TABLE Product (  
8     Id int not null identity(1,1) primary key,  
9     Name nvarchar(50) not null,  
10    Description nvarchar(50) not null,  
11    Price money not null  
12 )  
13  
14  
15 CREATE TABLE [Order] (  
16     Id int not null identity(1,1) primary key,  
17     CustomerId int not null references Customer(Id),  
18     OrderDate date not null,  
19     DueDate date not null,  
20     BillingTerms nvarchar(50) not null,  
21     ShippingTerms nvarchar(50) not null  
22 )  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```

117 % No issues found Ln: 32 Ch: 2 TABS CRLF

T-SQL Message

Command(s) completed successfully.

117 % No issues found Ln: 2 Ch: 1 MIXED MIXED

Toolbox Server Explorer SQL Server Obj... Query executed successfully at 18:08:25

Package Manager Console Output Web Publish Activity

Ready Add to Source Control

Properties Solution Explorer Team Explorer

Updater on tables

